UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P O Box 1450
Alexandria, Virginia 22313-1450
www uspto gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/647,058	08/21/2003	J. Patrick Thompson	MSFT-1748/302722.01	1588

41505 7590 05/28/2008
WOODCOCK WASHBURN LLP (MICROSOFT CORPORATION)
CIRA CENTRE, 12TH FLOOR
2929 ARCH STREET
PHILADELPHIA, PA 19104-2891

EXAMINER	
PHAM, MICHAEL	

ART UNIT	PAPER NUMBER
2167	

MAIL DATE	DELIVERY MODE
05/28/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	**Application No.** 10/647,058	**Applicant(s)** THOMPSON ET AL.	
	Examiner MICHAEL D. PHAM	**Art Unit** 2167	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *15 February 2008*.

2a)□ This action is **FINAL**. 2b)☒ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-11,13-21,23-25,27-31 and 33-60* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-11, 13-21, 23-25, 27-31, and 33-60* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)□ The drawing(s) filed on _____ is/are: a)□ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) □ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application

6) □ Other: _____.

DETAILED ACTION

1. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 2/15/08 has been entered.

Status of claims

2. Claims 1-11, 13-21, 23-25, 27-31, and 33-60 are pending.

3. Claims 1-11, 13-21, 23-25, 27-31, and 33-60 were examined.

Claim Objections

4. Claims 1, 21, and 37 are objected to because of the following informalities: missing semi-colon in claim 1 line 4; claim 21 line 5 double comma; and claim 37 line 12, recites can can. Appropriate correction is required.

Claim Rejections - 35 USC § 112

5. The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

6. Claims 1-11, 13-21-25, 27-31, and 33-60 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention. Claim 1 recites the following terms and phrases, "categories are themselves items" and "wherein any modifications to the specific Item in a specific Item Folder of the subset are reflected in each Item Folder of the subset" ; however these limitation do not appear to be supported by the specifications nor any of the original claims. Claims 21, 37, 43, 49, and 55 recite similar limitations. All other claims are rejected for failing to resolve the deficiencies of claims from which they depend.

7. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

8. Claims 49-54 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. Applicant's claim states that it is "…operable to be a member…" in claim 49 line 12. It is unclear what Applicant's intended metes and bounds of the claim are. All other claims are rejected for failing to resolve the deficiencies of claim 49.

9. Claims 21-25, 27-31, and 33-36 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which

applicant regards as the invention. Applicant's claim states that the claim is "...capable of..." in claim 21 line 1. It is unclear what Applicant's intended metes and bounds of the claim. All other claims are rejected for failing to resolve the deficiencies of claim 21.

10. Claims 43-60 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. In particular, claims 43, 49, and 55 recite mediums containing "instructions for"; however this does not limit the claim. It is unclear what Applicant's metes and bounds of the claim are. All other claims are rejected for failing to resolve the deficiencies of claim 43, 49, and 55.

11. Claim 55 recites the limitation "said operating system" in line 2. There is insufficient antecedent basis for this limitation in the claim.

12. Prior rejections on 11/15/07 under 112 are respectfully withdrawn.

Claim Rejections - 35 USC § 101

13. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

14. Claims 1—11, 13-21, 23-36, 43-48, 49-54, and 55-60 are rejected under 35 U.S.C. 101 because the language of the claims raise a question as to whether the claims are directed merely

to an abstract idea that is not tied to a machine which would result in a practical application to produce a concrete, useful, and tangible result to form the basis of statutory subject matter under 35 U.S.C. 101.

a. Claims 1 is rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

MPEP 2106.01:

The claims lack the necessary physical articles or objects to constitute a machine or a manufacture within the meaning of 35 USC 101. They are clearly not a series of steps or acts to be a process nor are they a combination of chemical compounds to be a composition of matter. As such, they fail to fall within a statutory category. They are, at best, functional descriptive material *per se.*

Descriptive material can be characterized as either "functional descriptive material" or "nonfunctional descriptive material." Both types of "descriptive material" are nonstatutory when claimed as descriptive material *per se*, 33 F.3d at 1360, 31 USPQ2d at 1759. When functional descriptive material is recorded on some computer-readable medium, it becomes structurally and functionally interrelated to the medium and will be statutory in most cases since use of technology permits the function of the descriptive material to be realized. Compare *In re Lowry*, 32 F.3d 1579, 1583-84, 32 USPQ2d 1031, 1035 (Fed. Cir. 1994)

Merely claiming nonfunctional descriptive material, i.e., abstract ideas, stored on a computer-readable medium, in a computer, or on an electromagnetic carrier signal, does not make it statutory. See *Diehr*, 450 U.S. at 185-86, 209 USPQ at 8 (noting that the claims for an algorithm in *Benson* were unpatentable as abstract ideas because "[t]he sole practical application of the algorithm was in connection with the programming of a general purpose computer.").

Claim 1 recites "a computer system". However claim 1 fails to contain any computer hardware that is used to implement the system so as to realize the functionality. Contrary to arguments made by some applicants, use of the word "system" does not inherently means that the claim is directed to a machine. Only if at least one of the claimed elements of the system is a physical part of a device can the system as claimed constitute part of a device or a combination of devices to be a machine within the meaning of 101. Furthermore, 0072, of US20050055354 (publication of application), it states the term "computer system" is intended to encompass any and all devices capable of storing and processing information and/or capable of using the stored information to control the behavior or execution of the device itself, regardless of whether such devices are electronic, mechanical, logical, or virtual in nature. In otherwords, the computer system appears to be software per se. Thus the body of claim 1 is merely an abstract idea and is being processed without any computer hardware.

b. Claims 21 is rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

MPEP 2106.01:

The claims lack the necessary physical articles or objects to constitute a machine or a manufacture within the meaning of 35 USC 101. They are clearly not a series of steps or acts to be a process nor are they a combination of chemical compounds to be a composition of matter. As such, they fail to fall within a statutory category. They are, at best, functional descriptive material *per se*.

Descriptive material can be characterized as either "functional descriptive material" or "nonfunctional descriptive material." Both types of "descriptive material" are nonstatutory when claimed as descriptive material *per se*, 33 F.3d at 1360, 31 USPQ2d at 1759. When functional descriptive material is recorded on some computer-readable medium, it becomes structurally and functionally interrelated to the medium and will be statutory in most cases since use of technology permits the function of the descriptive material to be realized. Compare *In re Lowry*, 32 F.3d 1579, 1583-84, 32 USPQ2d 1031, 1035 (Fed. Cir. 1994)

Merely claiming nonfunctional descriptive material, i.e., abstract ideas, stored on a computer-readable medium, in a computer, or on an electromagnetic carrier signal, does not make it statutory. See *Diehr*, 450 U.S. at 185-86, 209 USPQ at 8 (noting that the claims for an algorithm in *Benson* were unpatentable as abstract ideas because "[t]he sole practical application of the algorithm was in connection with the programming of a general purpose computer.").

Claim 21 recites “a hardware/software interface system". However claim 21 fails to contain any computer hardware that is used to implement the system so as to realize the functionality. Contrary to arguments made by some applicants, use of the word “system” does not inherently means that the claim is directed to a machine. Only if at least one of the claimed elements of the system is a physical part of a device can the system as claimed constitute part of a device or a combination of devices to be a machine within the meaning of 101. Furthermore, 0083, of US20050055354 (publication of application), it states that a hardware/software interface system is software, or a combination of hardware and software, that serves as the interface between the underlying hardware component of a computer system and applications that execute on the computer system. This appears to be software per se. Thus the body of claim 21 is merely an abstract idea and is being processed without any computer hardware.

c. Claims 43 is rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

Claim 43 recites a computer-readable medium. The claim fails to place the invention squarely within one statutory class of invention. On paragraph 1089, of US20050055354 (publication of application), applicant’s provide evidence that applicant intends the medium to include a server. A server in and of itself is a program. Accordingly, the claim is directed to a program itself, not a process occurring as a result of executing the program, a machine programmed to operate in accordance with the program nor a manufacture structurally and functionally interconnected with the program in a manner which enables the program to act as a computer component and realize its functionality.

d. Claims 49 is rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

Claim 49 recites a computer-readable medium. The claim fails to place the invention squarely within one statutory class of invention. On paragraph 1089, of US20050055354 (publication of application), applicant's provide evidence that applicant intends the medium to include a server. A server in and of itself is a program. Accordingly, the claim is directed to a program itself, not a process occurring as a result of executing the program, a machine programmed to operate in accordance with the program nor a manufacture structurally and functionally interconnected with the program in a manner which enables the program to act as a computer component and realize its functionality.

e. Claims 55 is rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

Claim 55 recites a computer-readable medium. The claim fails to place the invention squarely within one statutory class of invention. On paragraph 1089, of US20050055354 (publication of application), applicant's provide evidence that applicant intends the medium to include a server. A server in and of itself is a program. Accordingly, the claim is directed to a program itself, not a process occurring as a result of executing the program, a machine programmed to operate in accordance with the program nor a manufacture structurally and functionally interconnected with the program in a manner which enables the program to act as a computer component and realize its functionality.

15. All other claims fail to resolve the deficiencies of the claims from which they depend from, and are therefore rejected.

Claim Rejections - 35 USC § 103

16. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

17. Claims 1, 2, 3, 5, 7, 8, 10, 21, 37, 38, 39, 41, 43, 44, 45, 47, 49, 50, 51, 53, 55, 56, 57, and 59 are rejected under 35 U.S.C. 103(a) as being unpatentable over US patent application 2004/0199521 by Anglin et. al. (hereafter Anglin) further in view of US Patent 7158962 by Nelson (hereafter Nelson) and US Patent 6571245 by Huang et. al. (hereafter Huang).

<u>Claim 1:</u>

Regarding claim 1, Anglin discloses the following claimed limitations:

"a hardware/software interface system configured to manage a plurality of Items stored in a data store, wherein each of said plurality of Items constitute a discrete storable unit of information"[figure 1 a computing environment. 0024, storage objects may be defined as part of a storage group by adding the group identified of the one or more groups in which the storage object is a member to the group field of the storage object entry. 0020, fig. 2a illustrates the data structure of a storage object entry or record in the storage database that is added whenever a storage object is confirmed as written to the storage. The entry includes a unique identifier that uniquely identifies the storage object and entry in the storage database. Accordingly, disclosing a hardware/software interface system (fig 1) configured to manage a plurality of Items (storage objects) stored in a data store (database), wherein each of said plurality of Items (storage objects) constitute a discrete storable unit of information (fig. 2a)]

"the hardware/software interface system further configured to manage a plurality of Item Folders that constitute an organizational structure for said Items, and each Item Folder includes membership information identifying any relationship with Items;"[fig. 1 a computing environment. 0024, storage objects may be defined as part of a storage group by adding the group identified of the one or more groups in which the storage object is a member to the group field of the storage object entry. 0024, furthermore for every group to which the object belongs, the member IDs, field is updated for the group entry. 0021, the member Ids of any storage objects that are members of the group identified in field. Accordingly, disclosing the hardware/software interface system further configured to manage a plurality of Item Folders (storage group) that constitute an organizational structure for said Items (storage object), and each Item Folder (storage group) includes membership information identifying (member id field of the storage object) any relationship with Items (storage object).]

"the hardware/software interface system further configured to generate relationships between a specific Item and a subset of Items Folders of the plurality of Item Folders,'[0020, the field 58 indicates one or more groups to which the storage objects belongs. The field 58 may specify any number of groups i.e. zero or more, by listing one or more groups or having a pointer to a data structure indicating multiple groups with which the current object is associated. Accordingly, disclosing the hardware/software interface system further configured to generate relationships between a specific Item (storage object) and a subset of Items Folders (zero or more groups) of the plurality of Item Folders (groups)]

Anglin does not explicitly disclose "wherein the Item Folders are themselves Items" , "a plurality of Categories that constitute an additional organizational structure for said Items, at least one of said Items belonging to at least one of the Categories, wherein each Item in a specific Category includes a common attribute that is described for that specific Category, the Item Folders and the Categories arranged in a directed graph structure, and Categories are themselves Items;", and "the hardware/software interface system configured to render a user interface that displays the specific Item in each Item Folder of the subset,"

On the other hand, Nelson discloses col. 3 lines 59-60, an item can be for example a folder or a document. Accordingly, disclosing wherein the Item Folders are themselves Items (Items can be for example folder or a document).

Nelson further discloses, col. 1 lines 30-33, foldering or linking is a process where a content management system manages and/or controls the creation, retrieval, editing, and distribution of content within an information system. Foldering or liking is the act of forming a link relationship between two Items. Col. 1 lines 38-42, this relationship is maintained in a table known as a links table. Col. 1 lines 38-39, each item in the system may exist as the source and/or target for linking. Accordingly, disclosing a plurality of Categories (source/targets) that constitute an additional organizational structure for said Items (items), at least one of said Items belonging to at least one of the Categories (each item may exist as the source and/or target), wherein each Item in a specific Category includes a common attribute that is described for that specific Category (source/target), the Item Folders and the Categories arranged in a directed graph structure (figure 3A), and Categories are themselves Items (each item may exist as the source and/or target.).

Nelson further discloses in figure 3a,3b,and 3c hierarchy of folders displayed. Accordingly disclosing the hardware/software interface system configured to render a user interface that displays the specific Item in each Item Folder of the subset (figure 3a, 3b, 3c)

Anglin and Nelson are all related to file manipulation and are therefore within the same field of endeavor. It would have been obvious to one of ordinary skill in the art at the time the invention was made to apply Nelson's disclosure above to the disclosure of Anglin in order to improve it by supporting multiple levels of folders and mutliple attributes. This allows for automating the tasks of linking items to folders and improving performance of multiple nested folders.

However, Anglin and Nelson do not explicitly disclose "wherein any modifications to the specific Item in a specific Item Folder of the subset are reflected in each Item Folder of the subset."

On the other hand, Huang discloses col. 11 lines 61-67, the sync folders on the desktop PCs contain files and folder to be maintained "in sync" with their duplicates on the network. The user is able to access and manipulate the items in the sync folder like any other folders. These items are functionally indistinguishable to the user except for the sync feature. Accordingly, disclosing wherein any modifications to the specific Item (manipulate the items) in a specific Item Folder (sync folder) of the subset are reflected in each Item Folder of the subset (maintained in sync).

Anglin, Nelson, and Huang are all directed to file manipulation, and are therefore within the same field of endeavor. It would have been obvious to a person of an ordinary skill at the time the invention was made to have applied the disclosure of Huang to the combination of Anglin and Nelson for the purpose of synchronizing files and folders together for the purpose of maintaining files remotely.

Claim 2:

Anglin discloses "wherein an Item is a member of an Item Folder but is not owned by said Item Folder, such that the deletion of said Item Folder does not automatically result in the deletion of said Item." (See page 3, paragraph [0028] "The secondary deletion ensure that a storage object and corresponding storage object entry are only removed if the storage object is not a member of any further groups after eliminating the relationship between the storage object and group i." This follows the request to delete the "group leader" which represents the group ID of the "storage group" or the "Item Folder" as in referred to in the claim.)

Claim 3:

Anglin discloses "wherein an Item is automatically deleted when it no longer belongs to any Item Folder." (See page 3, paragraph [0029] "If the target group is the only indicated group in the associated groups fields for the specified storage object entry, then the storage management server deletes the specified storage object entry from the storage database and deletes the identifier of the deleted storage object entry from the group entry for the target group.")

Claim 5:

Anglin discloses "wherein said Item is automatically deleted when it is a member of only one Item Folder and said Item Folder is deleted." (See page 3, paragraph [0028] "The secondary deletion ensure that a storage object and corresponding storage object entry are only removed if the storage object is not a member of any further groups after eliminating the relationship between the storage object and group i." This follows the request to delete the "group leader" which represents the group ID of the "storage group" or the "Item Folder" as in referred to in the claim.)

Claim 7:

Anglin discloses "wherein each Item is a member of at least one Item Folder but is not owned by said Item Folder, such that the deletion of said Item Folder does not automatically result in the deletion of an Item." (See page 3, paragraph [0028] "The secondary deletion ensure that a storage object and corresponding storage object entry are only removed if the storage object is not a member of any further groups after eliminating the relationship between the storage object and group i." This follows the request to delete the "group leader" which represents the group ID of the "storage group" or the "Item Folder" as in referred to in the claim.)

Claim 8:

Anglin discloses "wherein each said Item is itself automatically deleted when it no longer belongs to any Item Folder." (See page 3, paragraph [0029] "If the target group is the only indicated group in the associated groups fields for the specified storage object entry, then the storage management server deletes the specified storage object entry from the storage database and deletes the identifier of the deleted storage object entry from the group entry for the target group.")

Claim 10:

Anglin discloses "wherein each said Item is itself automatically deleted when it is a member of only one Item Folder and said Item Folder is deleted." (See page 3, paragraph [0028] "The secondary deletion ensure that a storage object and corresponding storage object entry are only removed if the storage object is not a member of any further groups after eliminating the relationship between the storage object and group i." This follows the request to delete the "group leader" which represents the group ID of the "storage group" or the "Item Folder" as in referred to in the claim.)

Claim 21:

Regarding claim 21, Anglin discloses the following claimed limitations:

"manipulating an Item of a plurality of Items, said Item comprising a discrete unit of information comprising a basic set of properties commonly supported across objects exposed by an operating system shell, said Item being a fundamental unit of information manipulated by an operating system,"[figure 1 a computing environment. 0024, storage objects may be defined as part of a storage group by adding the group identified of the one or more groups in which the storage object is a member to the group field of the storage object entry. 0020, fig. 2a illustrates the data structure of a storage object entry or record in the storage database that is added whenever a storage object is confirmed as written to the storage. The entry includes a unique identifier that uniquely identifies the storage object and entry in the storage database. 0018, the server includes a storage management server program capable of performing storage related operations of data objects received from data management client programs. The storage management operations may comprise backup operations, achrival operations, hierarchical storage management related operations or any type of storage management operations known in the art. Accordingly, disclosing manipulating an Item of a plurality of Items (operations of data objects), said Item (storage object) comprising a discrete unit of information (figure 2a) comprising a basic set of properties commonly supported across objects (storage object entry) exposed by an operating system shell (storage management), said Item being a fundamental unit of information manipulated by an operating system (figure 2a)]

"and the hardware/software interface system is further configured to generate relationships between a specific Item and a subset of Item Folders of the plurality of Item Folders,"[fig. 1 a computing environment. 0024, storage objects may be defined as part of a storage group by adding the group identified of the one or more groups in which the storage object is a member to the group field of the storage object entry. 0024, furthermore for every group to which the object belongs, the member IDs, field is updated for the group entry. 0021, the member Ids of any storage objects that are members of the group identified in field. Accordingly, the hardware/software interface system is further configured to generate relationships (member IDs) between a specific Item (storage object) and a subset of Item Folders (storage group) of the plurality of Item Folders (storage groups).]

Anglin does not explicitly disclose "wherein the Item Folders are themselves Items" , "and said Item configured to be a member of one of a plurality of Categories that constitute an organizational structure for said Items, the Item Folders and the Categories arranged in a directed graph structure,", and "the hardware/software interface system configured to render a user interface that displays the specific Item in each Item Folder of the subset".

On the other hand, Nelson discloses col. 3 lines 59-60, an item can be for example a folder or a document. Accordingly, disclosing wherein said Item Folders are themselves Items (Items can be for example folder or a document).

Nelson further discloses, col. 1 lines 30-33, foldering or linking is a process where a content management system manages and/or controls the creation, retrieval, editing, and distribution of content within an information system. Foldering or liking is the act of forming a link relationship between two Items. Col. 1 lines 38-42, this relationship is maintained in a table known as a links table. Col. 1 lines 38-39, each item in the system may exist as the source and/or target for linking. Accordingly, disclosing and said Item configured to be a member of one of a plurality of Categories (source and/or target) that constitute an organizational structure for said Items (Items), the Item Folders and the Categories arranged in a directed graph structure (Figure 3A) and the hardware/software interface system configured to render a user interface that displays the specific Item in each Item Folder of the subset (figure 3a, 3b, 3c).

Anglin and Nelson are all related to file manipulation and are therefore within the same field of endeavor. It would have been obvious to one of ordinary skill in the art at the time the invention was made to apply Nelson's disclosure above to the disclosure of Anglin in order to improve it by supporting multiple levels of folders and mutliple attributes. This allows for automating the tasks of linking items to folders and improving performance of multiple nested folders.

However, Anglin and Nelson do not explicitly disclose "wherein any modifications to the specific Item in a specific Item Folder of the subset are reflected in each Item Folder of the subset," On the other hand, Huang discloses col. 11 lines 61-67, the sync folders on the desktop PCs contain files and folder to be maintained "in sync" with their duplicates on the network. The user is able to access and manipulate the items in the sync folder like any other folders. These items are functionally indistinguishable to the user except for the sync feature. Accordingly, disclosing wherein any modifications to the specific Item (manipulate the items) in a specific Item Folder (sync folder) of the subset are reflected in each Item Folder of the subset (maintained in sync).

Anglin, Nelson, and Huang are all directed to file manipulation, and are therefore within the same field of endeavor. It would have been obvious to a person of an ordinary skill at the time the invention was made to have applied the disclosure of Huang to the combination of Anglin and Nelson for the purpose of synchronizing files and folders together for the purpose of maintaining files remotely.

Claim 37:

Regarding claim 37, Anglin discloses the following claimed limitations:

"managing, by a hardware/software interface system, a plurality of Items stored in a data store, wherein each Item of said plurality of items constitutes discrete unit of information;"[figure 1 a computing environment. 0024, storage objects may be defined as part of a storage group by adding the group identified of the one or more groups in which the storage object is a member to the group field of the storage object entry. 0020, fig. 2a illustrates the data structure of a storage object entry or record in the storage database that is added whenever a storage object is confirmed as written to the storage. The entry includes a unique identifier that uniquely identifies the storage object and entry in the storage database. Accordingly, managing, by a hardware/software interface system, a plurality of Items (storage objects) stored in a data store (database), wherein each Item of said plurality of items constitutes discrete unit of information (figure 2a)]

"managing, by the hardware/software interface system, a plurality of Item Folders that constitute an organizational structure for said Items," [fig. 1 a computing environment. 0024, storage objects may be defined as part of a storage group by adding the group identified of the one or more groups in which the storage object is a member to the group field of the storage object entry. 0024, furthermore for every group to which the object belongs, the member IDs, field is updated for the group entry. 0021, the member Ids of any storage objects that are members of the group identified in field. Accordingly, disclosing managing, by the hardware/software interface system, a plurality of Item Folders (storage groups) that constitute an organizational structure for said Items (storage object).]

"each Item of the plurality can be a member of at least two Item Folders but is not owned by any of said Item Folders such that the deletion of any of said Item Folders does not automatically result in the deletion of said Item" [0028, the secondary deletion ensure that a storage object and corresponding storage object entry are only removed if the storage object is not a member of any further groups after eliminating the relationship between the storage object and group i. Accordingly, each Item of the plurality can be a member of at least two Item

Folders (groups) but is not owned by any of said Item Folders such that the deletion (deletion) of any of said Item Folders (eliminating the relationship between the storage object and group) does not automatically result in the deletion of said Item (only if the storage object is not a member of any further groups)]

" generating relationships between a specific Item and a subset of Item Folders in the plurality of Item Folders; and,'[0020, the field 58 indicates one or more groups to which the storage objects belongs. The field 58 may specify any number of groups i.e. zero or more, by listing one or more groups or having a pointer to a data structure indicating multiple groups with which the current object is associated. Accordingly, generating relationships between a specific Item (storage object) and a subset of Items Folders (zero or more groups) of the plurality of Item Folders (groups)]

Anglin does not explicitly disclose "wherein the Item Folders are themselves Items" , "each Item of the plurality can can be a member of a Category of a plurality of Categories that constitute an organizational structure for said Items, the Items Folders and the Categories arranged in a directed graph structure", and "rendering a user interface that displays the specific Item in each Item Folders of the subset,"

On the other hand, Nelson discloses col. 3 lines 59-60, an item can be for example a folder or a document. Accordingly, disclosing wherein the Item Folders are themselves Items (Items can be for example folder or a document).

Nelson further discloses, col. 1 lines 30-33, foldering or linking is a process where a content management system manages and/or controls the creation, retrieval, editing, and distribution of content within an information system. Foldering or liking is the act of forming a link relationship between two Items. Col. 1 lines 38-42, this relationship is maintained in a table known as a links table. Col. 1 lines 38-39, each item in the system may exist as the source and/or target for linking. Nelson further discloses in figure 3a,3b,and 3c hierarchy of folders displayed. Accordingly, each Item (item) of the plurality can can be a member of a Category of a plurality of Categories (source/target) that constitute an organizational structure for said Items (source/target), the Items Folders and the Categories arranged in a directed graph structure (figure 3a) and rendering a user interface that displays the specific Item in each Item Folders of the subset (figure 3a, 3b, 3c).

Anglin and Nelson are all related to file manipulation and are therefore within the same field of endeavor. It would have been obvious to one of ordinary skill in the art at the time the invention was made to apply Nelson's disclosure above to the disclosure of Anglin in order to improve it by supporting multiple levels of folders and mutliple attributes. This allows for automating the tasks of linking items to folders and improving performance of multiple nested folders.

However, Anglin and Nelson do not explicitly disclose "wherein any modifications to the specific Item in a specific Item Folder of the subset are reflected in each Item Folder of the subset."

On the other hand, Huang discloses col. 11 lines 61-67, the sync folders on the desktop PCs contain files and folder to be maintained "in sync" with their duplicates on the network. The user is able to access and manipulate the items in the sync folder like any other folders. These items are functionally indistinguishable to the user except for the sync feature. Accordingly, disclosing wherein any modifications to the specific Item (manipulate the items) in a specific Item Folder (sync folder) of the subset are reflected in each Item Folder of the subset (maintained in sync).

Anglin, Nelson, and Huang are all directed to file manipulation, and are therefore within the same field of endeavor. It would have been obvious to a person of an ordinary skill at the time the invention was made to have applied the disclosure of Huang to the combination of Anglin and Nelson for the purpose of synchronizing files and folders together for the purpose of maintaining files remotely.

Claim 38:

Anglin discloses "wherein the Item is a member of an Item Folder but is not owned by said Item Folder, such that the deletion of said Item Folder does not automatically result in the deletion of said Item." (See page 3, paragraph [0028] "The secondary deletion ensure that a storage object and corresponding storage object entry are only removed if the storage object is not a member of any further groups after eliminating the relationship between the storage object and group i."

This follows the request to delete the “group leader” which represents the group ID of the “storage group” or the “Item Folder” as in referred to in the claim.)

Claim 39:

Anglin discloses “wherein the Item is automatically deleted when it no longer belongs to any Item Folder.” (See page 3, paragraph [0029] “If the target group is the only indicated group in the associated groups fields for the specified storage object entry, then the storage management server deletes the specified storage object entry from the storage database and deletes the identifier of the deleted storage object entry from the group entry for the target group.”)

Claim 41:

Anglin discloses “The method of claim 38 wherein the Item is automatically deleted when it is a member of only one Item Folder and said Item Folder is deleted.” (See page 3, paragraph [0028] “The secondary deletion ensure that a storage object and corresponding storage object entry are only removed if the storage object is not a member of any further groups after eliminating the relationship between the storage object and group i.” This follows the request to delete the “group leader” which represents the group ID of the “storage group” or the “Item Folder” as in referred to in the claim.)

Claim 43:

Regarding claim 43, Anglin discloses the following claimed limitations:

“instructions for managing a plurality of Items, said Item comprising a discrete unit of information”[figure 1 a computing environment. 0024, storage objects may be defined as part of a storage group by adding the group identified of the one or more groups in which the storage object is a member to the group field of the storage object entry. 0020, fig. 2a illustrates the data structure of a storage object entry or record in the storage database that is added whenever a storage object is confirmed as written to the storage. The entry includes a unique identifier that uniquely identifies the storage object and entry in the storage database. Accordingly, instructions for managing a plurality of Items (storage objects), said Item comprising a discrete storable unit of information (fig. 2a)]

“instructions for managing a plurality of Item Folders that constitute an organizational structure for said items, and each Item Folder includes membership information identifying any relationships with Items;”[fig. 1 a computing environment. 0024, storage objects may be defined as part of a storage group by adding the group identified of the one or more groups in which the storage object is a member to the group field of the storage object entry. 0024, furthermore for every group to which the object belongs, the member IDs, field is updated for the group entry. 0021, the member Ids of any storage objects that are members of the group identified in field. Accordingly, disclosing instructions for managing a plurality of Item Folders (storage group) that constitute an organizational structure for said Items (storage object), and each Item Folder (storage group) includes membership information identifying (member id field of the storage object) any relationship with Items (storage object).]

" instructions for generating relationships between a specific Item and a subset of Item Folders in the plurality of Item Folders, and,’[0020, the field 58 indicates one or more groups to which the storage objects belongs. The field 58 may specify any number of groups i.e. zero or more, by listing one or more groups or having a pointer to a data structure indicating multiple groups with which the current object is associated. Accordingly, disclosing instructions for generating relationships between a specific Item (storage object) and a subset of Items Folders (zero or more groups) of the plurality of Item Folders (groups)]

Anglin does not explicitly disclose "wherein the Item Folders are themselves Items" , "instructions for managing a plurality of Categories that constitute an organizational structure for said Items, the categories arranged in a directed graph structure;", and "instructions for rendering a user interface that displays the specific Item in each Item Folder of the subset,,"

On the other hand, Nelson discloses col. 3 lines 59-60, an item can be for example a folder or a document. Accordingly, disclosing wherein the Item Folders are themselves Items (Items can be for example folder or a document).

Nelson further discloses, col. 1 lines 30-33, foldering or linking is a process where a content management system manages and/or controls the creation, retrieval, editing, and distribution of content within an information system. Foldering or liking is the act of forming a link relationship between two Items. Col. 1 lines 38-42, this relationship is maintained in a table known as a links table. Col. 1 lines 38-39, each item in the system may exist as the source and/or target for linking. Accordingly, disclosing a plurality of Categories (source/targets) that constitute an additional organizational structure for said Items (items), at least one of said Items belonging to at least one of the Categories (each item may exist as the source and/or target), wherein each Item in a specific Category includes a common attribute that is described for that specific Category (source/target), the Item Folders and the Categories arranged in a directed graph structure (figure 3A), and Categories are themselves Items (each item may exist as the source and/or target.).

Nelson further discloses in figure 3a,3b,and 3c hierarchy of folders displayed. Accordingly disclosing instructions for rendering a user interface that displays the specific Item in each Item Folder of the subset, (figure 3a, 3b, 3c).

Anglin and Nelson are all related to file manipulation and are therefore within the same field of endeavor. It would have been obvious to one of ordinary skill in the art at the time the invention was made to apply Nelson's disclosure above to the disclosure of Anglin in order to improve it by supporting multiple levels of folders and mutliple attributes. This allows for automating the tasks of linking items to folders and improving performance of multiple nested folders.

However, Anglin and Nelson do not explicitly disclose "wherein any modifications to the specific Item in a specific Item Folder of the subset are reflected in each Item Folder of the subset."

On the other hand, Huang discloses col. 11 lines 61-67, the sync folders on the desktop PCs contain files and folder to be maintained "in sync" with their duplicates on the network. The user is able to access and manipulate the items in the sync folder like any other folders. These items are functionally indistinguishable to the user except for the sync feature. Accordingly, disclosing wherein any modifications to the specific Item (manipulate the items) in a specific Item Folder (sync folder) of the subset are reflected in each Item Folder of the subset (maintained in sync).

Anglin, Nelson, and Huang are all directed to file manipulation, and are therefore within the same field of endeavor. It would have been obvious to a person of an ordinary skill at the time the invention was made to have applied the disclosure of Huang to the combination of Anglin and Nelson for the purpose of synchronizing files and folders together for the purpose of maintaining files remotely.

Claim 44:

Anglin discloses "wherein the Item is a member of an Item Folder but is not owned by said Item Folder, such that the deletion of said Item Folder does not automatically result in the deletion of said Item." (See page 3, paragraph [0028] "The secondary deletion ensure that a storage object and corresponding storage object entry are only removed if the storage object is not a member of any further groups after eliminating the relationship between the storage object and group i." This follows the request to delete the "group leader" which represents the group ID of the "storage group" or the "Item Folder" as in referred to in the claim.)

Claim 45:

Anglin discloses "The computer-readable medium of claim 44 wherein the Item is automatically deleted when it no longer belongs to any Item Folder" (See page 3, paragraph [0029] "If the target group is the only indicated group in the associated groups fields for the specified storage object entry, then the storage management server deletes the specified storage object entry from the storage database and deletes the identifier of the deleted storage object entry from the group entry for the target group.").

Claim 47:

Anglin discloses, "The computer-readable medium of claim 44 wherein the Item is automatically deleted when it is a member of only one Item Folder and said Item Folder is deleted." (See page 3, paragraph [0028] "The secondary deletion ensure that a storage object and corresponding storage object entry are only removed if the storage object is not a member of any further groups after eliminating the relationship between the storage object and group i." This follows the request to delete the "group leader" which represents the group ID of the "storage group" or the "Item Folder" as in referred to in the claim.)

Claim 49:

Regarding claim 43, Anglin discloses the following claimed limitations:

"instructions for managing, by a hardware/software interface system, a plurality of Items stored in a repository"[figure 1 a computing environment. 0024, storage objects may be defined as part of a storage group by adding the group identified of the one or more groups in which the storage object is a member to the group field of the storage object entry. 0020, fig. 2a illustrates the data structure of a storage object entry or record in the storage database that is added whenever a storage object is confirmed as written to the storage. The entry includes a unique identifier that uniquely identifies the storage object and entry in the storage database. Accordingly, instructions for managing, by a hardware/software interface system, a plurality of Items (storage objects) stored in a repository (database)]

"instructions for managing, by the hardware/software interface system, a plurality of Item Folders that constitute an organizational structure for said Items," [fig. 1 a computing environment. 0024, storage objects may be defined as part of a storage group by adding the group identified of the one or more groups in which the storage object is a member to the group field of the storage object entry. 0024, furthermore for every group to which the object belongs, the member IDs, field is updated for the group entry. 0021, the member Ids of any storage objects that are members of the group identified in field. Accordingly, disclosing instructions for managing a plurality of Item Folders (storage group) that constitute an organizational structure for said Items (storage object), and each Item Folder (storage group) includes membership information identifying (member id field of the storage object) any relationship with Items (storage object).]

"each Item of the plurality is configured to be a member of at least two Item Folders but is not owned by any of said Item Folders such that the deletion of said Item Folders does not automatically result in deletion of said Item" [0028, the secondary deletion ensure that a storage object and corresponding storage object entry are only removed if the storage object is not a member of any further groups after eliminating the relationship between the storage object and group i. Accordingly, each Item of the plurality is configured to be a member of at least two Item Folders (groups) but is not owned by any of said Item Folders such that the deletion (deletion) of any of said Item Folders (eliminating the relationship between the storage object and group) does not automatically result in the deletion of said Item (only if the storage object is not a member of any further groups)]

" instructions for generating relationships between a specific Item and a subset of Item Folders of the plurality of Item Folders, and"'[0020, the field 58 indicates one or more groups to which the storage objects belongs. The field 58 may specify any number of groups i.e. zero or more, by listing one or more groups or having a pointer to a data structure indicating multiple groups with which the current object is associated. Accordingly, disclosing instructions for generating relationships between a specific Item (storage object) and a subset of Items Folders (zero or more groups) of the plurality of Item Folders (groups)]

Anglin does not explicitly disclose “wherein the Item Folders are themselves Items” , “and each Item of the plurality is operable to be a member of a Category of a plurality of Categories that constitute an organizational structure for said Items, the Item Folders and the Categories arranged in a directed graph structure;", and “instructions for rendering a user interface that displays the specific Item in each Item Folder of the subset”

On the other hand, Nelson discloses col. 3 lines 59-60, an item can be for example a folder or a document. Accordingly, disclosing wherein the Item Folders are themselves Items (Items can be for example folder or a document).

Nelson further discloses, col. 1 lines 30-33, foldering or linking is a process where a content management system manages and/or controls the creation, retrieval, editing, and distribution of content within an information system. Foldering or liking is the act of forming a link relationship between two Items. Col. 1 lines 38-42, this relationship is maintained in a table known as a links table. Col. 1 lines 38-39, each item in the system may exist as the source and/or target for linking. Accordingly, disclosing and each Item (Item) of the plurality is operable to be a member of a Category of a plurality of Categories that constitute an organizational structure for said Items (sources and/or targets), the Item Folders and the Categories arranged in a directed graph structure (figure 3A).

Nelson further discloses in figure 3a,3b,and 3c hierarchy of folders displayed. Accordingly disclosing instructions for rendering a user interface that displays the specific Item in each Item Folder of the subset, (figure 3a, 3b, 3c).

Anglin and Nelson are all related to file manipulation and are therefore within the same field of endeavor. It would have been obvious to one of ordinary skill in the art at the time the invention was made to apply Nelson's disclosure above to the disclosure of Anglin in order to improve it by supporting multiple levels of folders and mutliple attributes. This allows for automating the tasks of linking items to folders and improving performance of multiple nested folders.

However, Anglin and Nelson do not explicitly disclose "wherein any modifications to the specific Item in a specific Item Folder of the subset are reflected in each Item Folder of the subset."

On the other hand, Huang discloses col. 11 lines 61-67, the sync folders on the desktop PCs contain files and folder to be maintained "in sync" with their duplicates on the network. The user is able to access and manipulate the items in the sync folder like any other folders. These items are functionally indistinguishable to the user except for the sync feature. Accordingly, disclosing wherein any modifications to the specific Item (manipulate the items) in a specific Item Folder (sync folder) of the subset are reflected in each Item Folder of the subset (maintained in sync).

Anglin, Nelson, and Huang are all directed to file manipulation, and are therefore within the same field of endeavor. It would have been obvious to a person of an ordinary skill at the time the invention was made to have applied the disclosure of Huang to the combination of Anglin and Nelson for the purpose of synchronizing files and folders together for the purpose of maintaining files remotely.

Claim 50:

Anglin discloses "The computer-readable medium of claim 49 wherein the Item is a member of an Item Folder but is not owned by said Item Folder, such that the deletion of said Item Folder does not automatically result in the deletion of said Item." (See page 3, paragraph [0028] "The secondary deletion ensure that a storage object and corresponding storage object entry are only removed if the storage object is not a member of any further groups after eliminating the relationship between the storage object and group i." This follows the request to delete the "group leader" which represents the group ID of the "storage group" or the "Item Folder" as in referred to in the claim.)

Claim 51:

Anglin discloses "The computer-readable medium of claim 50 wherein the Item is automatically deleted when it no longer belongs to any Item Folder." (See page 3, paragraph [0029] "If the target group is the only indicated group in the associated groups fields for the specified storage object entry, then the storage management server deletes the specified storage object entry from the storage database and deletes the identifier of the deleted storage object entry from the group entry for the target group.")

Claim 53:

Anglin discloses "The computer-readable medium of claim 50 wherein the Item is automatically deleted when it is a member of only one Item Folder and said Item Folder is deleted." (See page 3, paragraph [0028] "The secondary deletion ensure that a storage object and corresponding storage object entry are only removed if the storage object is not a member of any further groups after eliminating the relationship between the storage object and group i." This follows the request to delete the "group leader" which represents the group ID of the "storage group" or the "Item Folder" as in referred to in the claim.)

Claim 55:

Regarding claim 55, Anglin discloses the following claimed limitations:

"instructions for managing a plurality of Items stored in a repository, wherein each Item of said plurality of Items constitute a discrete unit of information that can be manipulated by a hardware/software interface system;"[figure 1 a computing environment. 0024, storage objects may be defined as part of a storage group by adding the group identified of the one or more groups in which the storage object is a member to the group field of the storage object entry. 0020, fig. 2a illustrates the data structure of a storage object entry or record in the storage database that is added whenever a storage object is confirmed as written to the storage. The entry includes a unique identifier that uniquely identifies the storage object and entry in the storage database. Accordingly, disclosing instructions for managing a plurality of Items (storage objects) stored in a repository (database), wherein each Item of said plurality of Items constitute a discrete unit of information (figure 2a) that can be manipulated by a hardware/software interface system (figure 1)]

"instructions for managing a plurality of Item Folders that constitute an organizational structure for said Items and each Item Folder includes membership information identifying any relationship with Items;"[fig. 1 a computing environment. 0024, storage objects may be defined as part of a storage group by adding the group identified of the one or more groups in which the storage object is a member to the group field of the storage object entry. 0024, furthermore for every group to which the object belongs, the member IDs, field is updated for the group entry. 0021, the member Ids of any storage objects that are members of the group identified in field. Accordingly, disclosing the hardware/software interface system further configured to manage a plurality of Item Folders (storage group) that constitute an organizational structure for said Items (storage object), and each Item Folder (storage group) includes membership information identifying (member id field of the storage object) any relationship with Items (storage object).]

"instructions for generating relationships between a specific item and a subset of item folders in the plurality of item folders," [0020, the field 58 indicates one or more groups to which the storage objects belongs. The field 58 may specify any number of groups i.e. zero or more, by listing one or more groups or having a pointer to a data structure indicating multiple groups with which the current object is associated. Accordingly, instructions for generating relationships between a specific Item (storage object) and a subset of Items Folders (zero or more groups) of the plurality of Item Folders (groups)]

Anglin does not explicitly disclose ", said Item Folders are themselves Items, wherein the Item Folders are themselves Items", "instructions for managing a plurality of Categories that constitute an additional organizational structure for said Items, at least one of said Items belonging to at least one of the Categories, the Item Folders and the Categories arranged in a directed graph structure,", and "instructions for rendering a user interface that displays the specific item in each item folder of the subset"

On the other hand, Nelson discloses col. 3 lines 59-60, an item can be for example a folder or a document. Accordingly, disclosing ", said Item Folders are themselves Items, wherein the Item Folders are themselves Items" (Items can be for example folder or a document).

Nelson further discloses, col. 1 lines 30-33, foldering or linking is a process where a content management system manages and/or controls the creation, retrieval, editing, and distribution of content within an information system. Foldering or liking is the act of forming a link relationship between two Items. Col. 1 lines 38-42, this relationship is maintained in a table known as a links table. Col. 1 lines 38-39, each item in the system may exist as the source and/or target for linking. Nelson further discloses in figure 3a,3b,and 3c hierarchy of folders displayed. Accordingly, disclosing instructions for managing a plurality of Categories that constitute an additional organizational structure for said Items (sources and/or target), at least one of said Items belonging to at least one of the Categories (sources and/or target), the Item Folders and the Categories arranged in a directed graph structure (figure 3a).

Nelson further discloses in figure 3a,3b,and 3c hierarchy of folders displayed. Accordingly disclosing instructions for rendering a user interface that displays the specific item in each item folder of the subset (figure 3a, 3b, 3c)

Anglin and Nelson are all related to file manipulation and are therefore within the same field of endeavor. It would have been obvious to one of ordinary skill in the art at the time the invention was made to apply Nelson's disclosure above to the disclosure of Anglin in order to improve it by supporting multiple levels of folders and mutliple attributes. This allows for automating the tasks of linking items to folders and improving performance of multiple nested folders.

However, Anglin and Nelson do not explicitly disclose “wherein any modifications to the specific Item in a specific Item Folder of the subset are reflected in each Item Folder of the subset.”

On the other hand, Huang discloses col. 11 lines 61-67, the sync folders on the desktop PCs contain files and folder to be maintained "in sync" with their duplicates on the network. The user is able to access and manipulate the items in the sync folder like any other folders. These items are functionally indistinguishable to the user except for the sync feature. Accordingly, disclosing wherein any modifications to the specific Item (manipulate the items) in a specific Item Folder (sync folder) of the subset are reflected in each Item Folder of the subset (maintained in sync).

Anglin, Nelson, and Huang are all directed to file manipulation, and are therefore within the same field of endeavor. It would have been obvious to a person of an ordinary skill at the time the invention was made to have applied the disclosure of Huang to the combination of Anglin and Nelson for the purpose of synchronizing files and folders together for the purpose of maintaining files remotely.

Claim 56:

Anglin discloses ”The computer-readable medium of claim 55 wherein an Item is a member of an Item Folder but is not owned by said Item Folder, such that the deletion of said Item Folder does not automatically result in the deletion of said Item.” (See page 3, paragraph [0028] “The secondary deletion ensure that a storage object and corresponding storage object entry are only removed if the storage object is not a member of any further groups after eliminating the relationship between the storage object and group i.” This follows the request to delete the “group leader” which represents the group ID of the “storage group” or the “Item Folder” as in referred to in the claim.)

Claim 57:

Anglin discloses “The computer-readable medium of claim 56 wherein the Item is automatically deleted when it no longer belongs to any Item Folder.” (See page 3, paragraph [0029] “If the target group is the only indicated group in the associated groups fields for the specified storage object entry, then the storage management server deletes the specified storage object entry from the storage database and deletes the identifier of the deleted storage object entry from the group entry for the target group.”)

Claim 59:

Anglin discloses ”The computer-readable medium of claim 56 wherein the Item is automatically deleted when it is a member of only one Item Folder and said Item Folder is deleted.” (See page

3, paragraph [0028] "The secondary deletion ensure that a storage object and corresponding storage object entry are only removed if the storage object is not a member of any further groups after eliminating the relationship between the storage object and group i." This follows the request to delete the "group leader" which represents the group ID of the "storage group" or the "Item Folder" as in referred to in the claim.)

18. Claims 4, 6, 9, 11, 40, 42, 46, 48, 52, 54, 58, and 60 rejected under 35 U.S.C. 103(a) as being unpatentable over US patent application 2004/0199521 by Anglin et. al. (hereafter Anglin), US Patent 7158962 by Nelson (hereafter Nelson), and US Patent 6571245 by Huang et. al. (hereafter Huang) further in view of US 20040073560 by Edwards (hereafter Edwards)

<u>Claim 4:</u>

Anglin, Nelson, and Huang do not explicitly teach an Item, when it no longer belongs to any Item Folder, automatically becomes a member of a default Item Folder. However, Edwards teaches more explicitly an Item, when it no longer belongs to any Item Folder, automatically becomes a member of a default Item Folder. (See page 3, paragraph [0038] "The synchroniser can be set to 'Recycle' rather than delete files. This means that whenever the synchroniser is to over-write or delete a file, the file is passed to the operating system to be 'recycled'. This means it is not deleted immediately, but stored in an area where it can be retrieved if required." In the instant application, the area in which the item is stored is called the default Item Folder). It would have been obvious to one with ordinary skill in the art to combine the Item and Item folder system of Anglin, Nelson, and Huang with the disclosure of a recycling method of Edwards by simply adding the recycling method of Edwards to the system of Anglin, Nelson, and Huang. Edwards points out the advantage of keeping the potentially deleted file (or in the instant application, the object,) available for future use rather than immediate deletion. It is for this reason that one of ordinary skill in the art would have been motivated to have an Item, when it no longer belongs to any Item Folder, automatically becomes a member of a default Item Folder.

Claim 6:

Anglin, Nelson, and Huang do not explicitly teach said Item, when it is a member of only one Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder. However, Edwards teaches said Item, when it is a member of only one Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder. (See page 3, paragraph [0038] "The synchroniser can be set to 'Recycle' rather than delete files. This means that whenever the synchroniser is to over-write or delete a file, the file is passed to the operating system to be 'recycled'. This means it is not deleted immediately, but stored in an area where it can be retrieved if required." In the instant application, the area in which the item is stored is called the default Item Folder.) It would have been obvious to one with ordinary skill in the art to combine the Item and Item folder system of Anglin, Nelson, and Huang with the disclosure of a recycling method of Edwards by simply adding the recycling method of Edwards to the system of Anglin, Nelson, and Huang. Edwards points out the advantage of keeping the potentially deleted file (or in the instant application, the object,) available for future use rather than immediate deletion. It is for this reason that one of ordinary skill in the art would have been motivated to have said Item, when it is a member of only one Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder.

Claim 9:

Anglin, Nelson, and Huang do no explicitly teach each said Item, when each no longer belongs to any Item Folder, automatically become members of a default Item Folder. Nelson does disclose col. 5 lines 54-56, No folder exists matching these attributes. Consequently, the library server 25 creates a folder with the following attributes. However, Edwards more explicitly teaches each said Item, when each no longer belongs to any Item Folder, automatically become members of a default Item Folder. (See page 3, paragraph [0038] "The synchroniser can be set to 'Recycle' rather than delete files. This means that whenever the synchroniser is to over-write or delete a file, the file is passed to the operating system to be 'recycled'. This means it is not deleted immediately, but stored in an area where it can be retrieved if required." In the instant application, the area in which the item is stored is called the default Item Folder.) It would have been obvious to one with ordinary skill in the art to combine the Item and Item folder system of Anglin, Nelson, and Huang with the disclosure of a recycling method of Edwards by simply adding the recycling method of Edwards to the system of Anglin, Nelson, and Huang. Edwards points out the advantage of keeping the potentially deleted file (or in the instant application, the object,) available for future use rather than immediate deletion. It is for this reason that one of ordinary skill in the art would have been motivated to have each said Item, when each no longer belongs to any Item Folder, automatically become members of a default Item Folder.

Claim 11:

Anglin, Nelson, and Huang do not explicitly teach each said Item, when each is a member of only one Item Folder and said Item Folder is deleted, automatically become members of a default Item Folder. Nelson does disclose col. 5 lines 54-56, No folder exists matching these attributes. Consequently, the library server 25 creates a folder with the following attributes. However, Edwards teaches each said Item, when each is a member of only one Item Folder and said Item Folder is deleted, automatically become members of a default Item Folder. (See page 3, paragraph [0038] "The synchroniser can be set to 'Recycle' rather than delete files. This means that whenever the synchroniser is to over-write or delete a file, the file is passed to the operating system to be 'recycled'. This means it is not deleted immediately, but stored in an area where it can be retrieved if required." In the instant application, the area in which the item is stored is called the default Item Folder.) It would have been obvious to one with ordinary skill in the art to combine the Item and Item folder system of Anglin, Nelson, and Huang with the disclosure of a recycling method of Edwards by simply adding the recycling method of Edwards to the system of Anglin, Nelson, and Huang. Edwards points out the advantage of keeping the potentially deleted file (or in the instant application, the object,) available for future use rather than immediate deletion. It is for this reason that one of ordinary skill in the art would have been motivated to have each said Item, when each is a member of only one Item Folder and said Item Folder is deleted, automatically become members of a default Item Folder.

Claim 40:

Anglin, Nelson, and Huang do not explicitly teach said Item, when it no longer belongs to any Item Folder, automatically becomes a member of a default Item Folder. However, Edwards teaches said Item, when it no longer belongs to any Item Folder, automatically becomes a member of a default Item Folder. (See page 3, paragraph [0038] "The synchroniser can be set to 'Recycle' rather than delete files. This means that whenever the synchroniser is to over-write or delete a file, the file is passed to the operating system to be 'recycled'. This means it is not deleted immediately, but stored in an area where it can be retrieved if required." In the instant application, the area in which the item is stored is called the default Item Folder). It would have been obvious to one with ordinary skill in the art to combine the Item and Item folder system of Anglin, Nelson, and Huang with the disclosure of a recycling method of Edwards by simply adding the recycling method of Edwards to the system of Anglin, Nelson, and Huang. Edwards points out the advantage of keeping the potentially deleted file (or in the instant application, the object,) available for future use rather than immediate deletion. It is for this reason that one of ordinary skill in the art would have been motivated to have said Item, when it no longer belongs to any Item Folder, automatically becomes a member of a default Item Folder.

Claim 42:

Anglin, Nelson, and Huang do not explicitly teach said Item, when it is a member of only one Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder. However, Edwards teaches said Item, when it is a member of only one Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder. (See page 3, paragraph [0038] “The synchroniser can be set to ‘Recycle’ rather than delete files. This means that whenever the synchroniser is to over-write or delete a file, the file is passed to the operating system to be ‘recycled’. This means it is not deleted immediately, but stored in an area where it can be retrieved if required.” In the instant application, the area in which the item is stored is called the default Item Folder.) It would have been obvious to one with ordinary skill in the art to combine the Item and Item folder system of Anglin, Nelson, and Huang with the disclosure of a recycling method of Edwards by simply adding the recycling method of Edwards to the system of Anglin, Nelson, and Huang. Edwards points out the advantage of keeping the potentially deleted file (or in the instant application, the object,) available for future use rather than immediate deletion. It is for this reason that one of ordinary skill in the art would have been motivated to have said Item, when it is a member of only one Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder.

Claim 46:

Anglin, Nelson, and Huang do not explicitly disclose “said Item, when it no longer belongs to any Item Folder, automatically becomes a member of a default Item Folder.“ However, Edwards teaches said Item, when it no longer belongs to any Item Folder, automatically becomes a member of a default Item Folder. (See page 3, paragraph [0038] “The synchroniser can be set to ‘Recycle’ rather than delete files. This means that whenever the synchroniser is to over-write or delete a file, the file is passed to the operating system to be 'recycled'. This means it is not deleted immediately, but stored in an area where it can be retrieved if required." In the instant application, the area in which the item is stored is called the default Item Folder) It would have been obvious to one with ordinary skill in the art to combine the Item and Item folder system of Anglin, Nelson, and Huang with the disclosure of a recycling method of Edwards by simply adding the recycling method of Edwards to the system of Anglin, Nelson, and Huang. Edwards points out the advantage of keeping the potentially deleted file (or in the instant application, the object,) available for future use rather than immediate deletion. It is for this reason that one of ordinary skill in the art would have been motivated to have said Item, when it no longer belongs to any Item Folder, automatically becomes a member of a default Item Folder.

Claim 48:

Anglin, Nelson, and Huang do not explicitly disclose "when it is a member of only one Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder." However, Edwards teaches said Item, when it is a member of only one Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder. (See page 3, paragraph [0038] "The synchroniser can be set to 'Recycle' rather than delete files. This means that whenever the synchroniser is to over-write or delete a file, the file is passed to the operating system to be 'recycled'. This means it is not deleted immediately, but stored in an area where it can be retrieved if required." In the instant application, the area in which the item is stored is called the default Item Folder.) It would have been obvious to one with ordinary skill in the art to combine the Item and Item folder system of Anglin, Nelson, and Huang with the disclosure of a recycling method of Edwards by simply adding the recycling method of Edwards to the system of Anglin, Nelson, and Huang. Edwards points out the advantage of keeping the potentially deleted file (or in the instant application, the object,) available for future use rather than immediate deletion. It is for this reason that one of ordinary skill in the art would have been motivated to have said Item, when it is a member of only one Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder.

Claim 52:

Anglin, Nelson, and Huang do not explicitly disclose "said Item, when it no longer belongs to any Item Folder, automatically becomes a member of a default Item Folder." However, Edwards teaches said Item, when it no longer belongs to any Item Folder, automatically becomes a member of a default Item Folder. (See page 3, paragraph [0038] "The synchroniser can be set to 'Recycle' rather than delete files. This means that whenever the synchroniser is to over-write or delete a file, the file is passed to the operating system to be 'recycled'. This means it is not deleted immediately, but stored in an area where it can be retrieved if required." In the instant application, the area in which the item is stored is called the default Item Folder) It would have been obvious to one with ordinary skill in the art to combine the Item and Item folder system of Anglin, Nelson, and Huang with the disclosure of a recycling method of Edwards by simply adding the recycling method of Edwards to the system of Anglin, Nelson, and Huang. Edwards points out the advantage of keeping the potentially deleted file (or in the instant application, the object,) available for future use rather than immediate deletion. It is for this reason that one of ordinary skill in the art would have been motivated to have said Item, when it no longer belongs to any Item Folder, automatically becomes a member of a default Item Folder.

Claim 54:

Anglin, Nelson, and Huang do not explicitly disclose "said Item, when it is a member of only one Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder." However, Edwards teaches said Item, when it is a member of only one Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder. (See page 3, paragraph [0038] "The synchroniser can be set to 'Recycle' rather than delete files. This means that whenever the synchroniser is to over-write or delete a file, the file is passed to the operating system to be 'recycled'. This means it is not deleted immediately, but stored in an area where it can be retrieved if required." In the instant application, the area in which the item is stored is called the default Item Folder.) It would have been obvious to one with ordinary skill in the art to combine the Item and Item folder system of Anglin, Nelson, and Huang with the disclosure of a recycling method of Edwards by simply adding the recycling method of Edwards to the system of Anglin, Nelson, and Huang. Edwards points out the advantage of keeping the potentially deleted file (or in the instant application, the object,) available for future use rather than immediate deletion. It is for this reason that one of ordinary skill in the art would have been motivated to have said Item, when it is a member of only one

Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder.

Claim 58:

Anglin, Nelson, and Huang do not explicitly disclose "wherein said Item, when it no longer belongs to any Item Folder, automatically becomes a member of a default Item Folder." However, Edwards teaches an Item, when it no longer belongs to any Item Folder, automatically becomes a member of a default Item Folder. (See page 3, paragraph [0038] "The synchroniser can be set to 'Recycle' rather than delete files. This means that whenever the synchroniser is to over-write or delete a file, the file is passed to the operating system to be 'recycled'. This means it is not deleted immediately, but stored in an area where it can be retrieved if required." In the instant application, the area in which the item is stored is called the default Item Folder) It would have been obvious to one with ordinary skill in the art to combine the Item and Item folder system of Anglin, Nelson, and Huang with the disclosure of a recycling method of Edwards by simply adding the recycling method of Edwards to the system of Anglin, Nelson, and Huang. Edwards points out the advantage of keeping the potentially deleted file (or in the instant application, the object,) available for future use rather than immediate deletion. It is for this reason that one of ordinary skill in the art would have been motivated to have an Item, when it no longer belongs to any Item Folder, automatically becomes a member of a default Item Folder.

Claim 60:

Anglin, Nelson, and Huang do not explicitly disclose "said Item, when it is a member of only one Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder." However, Edwards teaches said Item, when it is a member of only one Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder. (See page 3, paragraph [0038] "The synchroniser can be set to 'Recycle' rather than delete files. This means that whenever the synchroniser is to over-write or delete a file, the file is passed to the operating system to be 'recycled'. This means it is not deleted immediately, but stored in an area where it can be retrieved if required." In the instant application, the area in which the item is stored is called the default Item Folder.) It would have been obvious to one with ordinary skill in the art to combine the Item and Item folder system of Anglin, Nelson, and Huang with the disclosure of a recycling method of Edwards by simply adding the recycling method of Edwards to the system of Anglin, Nelson, and Huang. Edwards points out the advantage of keeping the potentially deleted file (or in the instant application, the object,) available for future use rather than immediate deletion. It is for this reason that one of ordinary skill in the art would have been motivated to have said Item, when it is a member of only one Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder.

19. Claims 13-20 and 33-36 are rejected under 35 U.S.C. 103(a) as being unpatentable over US patent application 2004/0199521 by Anglin et. al. (hereafter Anglin), US Patent

7158962 by Nelson (hereafter Nelson), and US Patent 6571245 by Huang et. al. (hereafter Huang) further in view of US Patent 6578046 by Chang et. al. (hereafter Chang)

Claim 13:

Anglin, Nelson, and Huang do not explicitly disclose "The computer system of claim 1 wherein a category is defined by an Item property." Nelson discloses col. 4 lines 60-62, automatic linking rules for that item type. These rules are stored in a separate table in the library server. However, Chang more explicitly teaches a Category is defined by an Item property. (See column 13, lines 26-30 "In the preferred embodiment, a FederatedCollection allows an application program to process data objects resulting from a query as a group or collection and at the same time preserves the sub-groupings relationships that exist between them." Simply by the Category being formed as the result of a query, the Item property is necessarily what defined the Category, as the data must meet the Item property in order to be a result of the query.) It would have been obvious to one with ordinary skill the art to combine the system as disclosed in Anglin, Nelson, and Huang with the disclosure of Chang because it would be logical to use the Item property as what defines the categories, especially in the case of a query. It is for this reason that one of ordinary skill in the art would have been motivated to have a Category is defined by an Item property.

Claim 14:

Anglin, Nelson, and Huang do not explicitly disclose "wherein one of said plurality of Categories is defined by an Item Property and only an Item comprising the Item property for a specific Category from among said plurality of Categories can be a member of said specific Category." The combination of Anglin, Nelson, and Huang disclose in Nelson col. 4 lines 60-62, automatic linking rules for that item type. These rules are stored in a separate table in the library server. However, Chang teaches more explicitly wherein one of said plurality of Categories is defined by an Item Property and only an Item comprising the Item property for a specific Category from among said plurality of Categories can be a member of said specific Category. (See column 13, lines 26-30 "In the preferred embodiment, a FederatedCollection allows an application program to process data objects resulting from a query as a group or collection and at the same time preserves the sub-groupings relationships that exist between them." Simply by the Category being formed as the result of a query, the Item property is necessarily what defined the Category, as the data must meet the Item property in order to be a result of the query. Also, by the nature of queries only returning the results that are related, the only members of the Category will be from results that comprise the Item property.) It would have been obvious to one with ordinary skill the art to combine the system as disclosed in Anglin, Nelson, and Huang with the disclosure of Chang because it would be logical to use the Item property as what defines the categories, especially in the case of a query. It is for this reason that one of ordinary skill in the art would have been motivated to have one of said plurality of Categories is defined by an Item property, and only an Item comprising the Item property for a specific Category from among said plurality of Categories can be a member of said specific Category.

Claim 15:

Anglin, Nelson, and Huang do not explicitly disclose an Item comprising the Item property for one of said plurality of Categories is automatically a member of that one of said plurality of Categories. However, Chang teaches an Item comprising the Item property for one of said plurality of Categories is automatically a member of that one of said plurality of Categories. (See column 13, lines 26-30 "In the preferred embodiment, a FederatedCollection allows an application program to process data objects resulting from a query as a group or collection and at the same time preserves the sub-groupings relationships that exist between them." By the nature of queries only returning the results that are related, the only members of the Category will be from results that comprise the Item property. Here, all of the results of the query are included in on the collection [referred to in the instant application as a category].) It would have been obvious to one with ordinary skill the art to combine the system as disclosed in Anglin, Nelson, and Huang with the disclosure of Chang because keeping the results of the query, all of which exhibit a relationship, is useful in that the query will not have to be run again. It is for this reason that one of ordinary skill in the art would have been motivated to include an Item comprising the Item property for one of said plurality of Categories is automatically a member of that one of said plurality of Categories.

Claim 16:

Anglin, Nelson, and Huang do not explicitly disclose an Item comprising one or more Item properties corresponding to one or more Categories of said plurality of Categories is automatically a member of each such Categories for said corresponding Item properties. However, Chang teaches an Item comprising one or more Item properties corresponding to one or more Categories of said plurality of Categories is automatically a member of each such Categories for said corresponding Item properties. (See column 13, lines 26-30 "In the preferred embodiment, a FederatedCollection allows an application program to process data objects resulting from a query as a group or collection and at the same time preserves the sub-groupings relationships that exist between them." By the nature of queries only returning the results that are related, the only members of the Category will be from results that comprise the Item property. Here, all of the results of the query are included in on the collection [referred to in the instant application as a category].) It would have been obvious to one with ordinary skill the art to combine the system as disclosed in Anglin, Nelson, and Huang with the disclosure of Chang because keeping the results of the query, all of which exhibit a relationship, is useful in that the query will not have to be run again. It is for this reason that one of ordinary skill in the art would have been motivated to include an Item comprising one or more Item properties corresponding to one or more Categories of said plurality of Categories is automatically a member of each such Categories for said corresponding Item properties.

Claim 17:

Anglin, Nelson, and Huang do not explicitly discloses "wherein each of said plurality of categories is defined by an item property." Nelson does disclose col. 4 lines 60-62, automatic linking rules for that item type. These rules are stored in a separate table in the library server. However, Chang more explicitly teaches each of said plurality of Categories is defined by an Item property. (See column 13, lines 26-30 "In the preferred embodiment, a FederatedCollection allows an application program to process data objects resulting from a query as a group or collection and at the same time preserves the sub-groupings relationships that exist between them." Simply by the Category being formed as the result of a query, the Item property is necessarily what defined the Category, as the data must meet the Item property in order to be a result of the query.) It would have been obvious to one with ordinary skill the art to combine the system as disclosed in Anglin, Nelson, and Huang with the disclosure of Chang because it would be logical to use the Item property as what defines the categories, especially in the case of a query. It is for this reason that one of ordinary skill in the art would have been motivated to have each of said plurality of Categories is defined by an Item property.

Claim 18:

Anglin, Nelson, and Huang do not explicitly teach each of said plurality of Categories is defined by an Item property, and only Items comprising the Item property for a specific Category from among said plurality of Categories can be members of said specific Category. However, Chang teaches each of said plurality of Categories is defined by an Item property, and only Items comprising the Item property for a specific Category from among said plurality of Categories can be members of said specific Category. (See column 13, lines 26-30 "In the preferred embodiment, a FederatedCollection allows an application program to process data objects resulting from a query as a group or collection and at the same time preserves the sub-groupings relationships that exist between them." Simply by the Category being formed as the result of a query, the Item property is necessarily what defined the Category, as the data must meet the Item property in order to be a result of the query. Also, by the nature of queries only returning the results that are related, the only members of the Category will be from results that comprise the Item property.) It would have been obvious to one with ordinary skill the art to combine the system as disclosed in Anglin, Nelson, and Huang with the disclosure of Chang because it would be logical to use the Item property as what defines the categories, especially in the case of a query. It is for this reason that one of ordinary skill in the art would have been motivated to have each of said plurality of Categories is defined by an Item property, and only Items comprising the Item property for a specific Category from among said plurality of Categories can be members of said specific Category.

<u>**Claim 19:**</u>

Anglin, Nelson, and Huang do not explicitly teach each Item comprising the Item property for one of said plurality of Categories are automatically members of that one of said plurality of Categories. However, Chang teaches each Item comprising the Item property for one of said plurality of Categories are automatically members of that one of said plurality of Categories. (See column 13, lines 26-30 "In the preferred embodiment, a FederatedCollection allows an application program to process data objects resulting from a query as a group or collection and at the same time preserves the sub-groupings relationships that exist between them." By the nature of queries only returning the results that are related, the only members of the Category will be from results that comprise the Item property. Here, all of the results of the query are included in on the collection [referred to in the instant application as a category].) It would have been obvious to one with ordinary skill the art to combine the system as disclosed in Anglin, Nelson, and Huang with the disclosure of Chang because keeping the results of the query, all of which exhibit a relationship, is useful in that the query will not have to be run again. It is for this reason that one of ordinary skill in the art would have been motivated to include each Item comprising the Item property for one of said plurality of Categories are automatically members of that one of said plurality of Categories.

Claim 20:

Anglin, Nelson, and Huang do not explicitly teach all Items comprising one or more Item properties corresponding to one or more Categories of said plurality of Categories are automatically members of all such Categories for said corresponding Item properties. However, Chang teaches all Items comprising one or more Item properties corresponding to one or more Categories of said plurality of Categories are automatically members of all such Categories for said corresponding Item properties. (See column 13, lines 26-30 "In the preferred embodiment, a FederatedCollection allows an application program to process data objects resulting from a query as a group or collection and at the same time preserves the sub-groupings relationships that exist between them." By the nature of queries only returning the results that are related, the only members of the Category will be from results that comprise the Item property. Here, all of the results of the query are included in on the collection [referred to in the instant application as a category].) It would have been obvious to one with ordinary skill the art to combine the system as disclosed in Anglin, Nelson, and Huang with the disclosure of Chang because keeping the results of the query, all of which exhibit a relationship, is useful in that the query will not have to be run again. It is for this reason that one of ordinary skill in the art would have been motivated to include all Items comprising one or more Item properties corresponding to one or more Categories of said plurality of Categories are automatically members of all such Categories for said corresponding Item properties.

Claim 33:

Anglin, Nelson, and Huang do not explicitly discloses "The hardware/software interface system of claim 21 wherein said category is defined by an item property". Nelson discloses col. 4 lines 60-62, automatic linking rules for that item type. These rules are stored in a separate table in the library server. However, Chang more explicitly teaches said Category is defined by an Item property. (See column 13, lines 26-30 "In the preferred embodiment, a FederatedCollection allows an application program to process data objects resulting from a query as a group or collection and at the same time preserves the sub-groupings relationships that exist between them." Simply by the Category being formed as the result of a query, the Item property is necessarily what defined the Category, as the data must meet the Item property in order to be a result of the query.) It would have been obvious to one with ordinary skill the art to combine the system as disclosed in Anglin, Nelson, and Huang with the disclosure of Chang because it would be logical to use the Item property as what defines the categories, especially in the case of a query. It is for this reason that one of ordinary skill in the art would have been motivated to have said Category is defined by an Item property.

Claim 34:

Anglin, Nelson, and Huang do not explicitly disclose said Category is defined by an Item property, and only an Item comprising the Item property for said Category can be a member of said Category. However, Chang teaches said Category is defined by an Item property, and only an Item comprising the Item property for said Category can be a member of said Category. (See column 13, lines 26-30 "In the preferred embodiment, a FederatedCollection allows an application program to process data objects resulting from a query as a group or collection and at the same time preserves the sub-groupings relationships that exist between them." Simply by the Category being formed as the result of a query, the Item property is necessarily what defined the Category, as the data must meet the Item property in order to be a result of the query. Also, by the nature of queries only returning the results that are related, the only members of the Category will be from results that comprise the Item property.) It would have been obvious to one with ordinary skill the art to combine the system as disclosed in Anglin, Nelson, and Huang with the disclosure of Chang because it would be logical to use the Item property as what defines the categories, especially in the case of a query. It is for this reason that one of ordinary skill in the art would have been motivated to have said Category is defined by an Item property, and only an Item comprising the Item property for said Category can be a member of said Category.

Claim 35:

Anglin, Nelson, and Huang do not explicitly disclose an Item comprising the Item property for one of said plurality of Categories is automatically a member of that one of said plurality of Categories. However, Chang teaches an Item comprising the Item property for one of said plurality of Categories is automatically a member of that one of said plurality of Categories. (See column 13, lines 26-30 "In the preferred embodiment, a FederatedCollection allows an application program to process data objects resulting from a query as a group or collection and at the same time preserves the sub-groupings relationships that exist between them." By the nature of queries only returning the results that are related, the only members of the Category will be from results that comprise the Item property. Here, all of the results of the query are included in on the collection [referred to in the instant application as a category].) It would have been obvious to one with ordinary skill the art to combine the system as disclosed in Anglin, Nelson, and Huang with the disclosure of Chang because keeping the results of the query, all of which exhibit a relationship, is useful in that the query will not have to be run again. It is for this reason that one of ordinary skill in the art would have been motivated to include an Item comprising the Item property for one of said plurality of Categories is automatically a member of that one of said plurality of Categories.

Claim 36:

Anglin, Nelson, and Huang do not explicitly disclose an Item comprising one or more Item properties corresponding to one or more Categories is automatically a member of each such Categories having at least one of said corresponding Item properties. However, Chang teaches an Item comprising one or more Item properties corresponding to one or more Categories is automatically a member of each such Categories having at least one of said corresponding Item properties. (See column 13, lines 26-30 "In the preferred embodiment, a FederatedCollection allows an application program to process data objects resulting from a query as a group or collection and at the same time preserves the sub-groupings relationships that exist between them." By the nature of queries only returning the results that are related, the only members of the Category will be from results that comprise the Item property. Here, all of the results of the query are included in on the collection [referred to in the instant application as a category].) It would have been obvious to one with ordinary skill the art to combine the system as disclosed in Anglin, Nelson, and Huang with the disclosure of Chang because keeping the results of the query, all of which exhibit a relationship, is useful in that the query will not have to be run again. It is for this reason that one of ordinary skill in the art would have been motivated to include an Item comprising one or more Item properties corresponding to one or more Categories is automatically a member of each such Categories having at least one of said corresponding Item properties.

20. Claims 23, 27, 28, and 30 rejected under 35 U.S.C. 103(a) as being unpatentable over US patent application 2004/0199521 by Anglin et. al. (hereafter Anglin), US Patent 7158962 by Nelson (hereafter Nelson), and US Patent 6571245 by Huang et. al. (hereafter Huang) further in view of US Patent 6438545 by Beauregard et. al. (hereafter Beauregard)

Claim 23:

Anglin, Nelson, and Huang do not explicitly disclose "wherein said Item is a fundamental unit of information manipulated by a virtual machine manager". However, Beauregard teaches said Item is a fundamental unit of information manipulated by a virtual machine manager. (See column 13, lines 12-16 "This broad I/O capability can be provided under the Virtual Machine Manager (VMM) that is available under Win32. The VMM is an extensible operating system whose core and standard components are provided by Microsoft Corporation.") Because of the advantages provided by VMM as taught in Beauregard, such as the broad I/O capability, it would have been obvious to one with ordinary skill in the art to combine the VMM of Beauregard with the teaching of Anglin, Nelson, and Huang. It is for this reason that one of ordinary skill in the art would have been motivated to have said Item is a fundamental unit of information manipulated by a virtual machine manager.

Claim 27:

Anglin discloses "The hardware/software interface system of claim 23 wherein said Item is not owned by said Item Folder, such that the deletion of said Item Folder does not automatically result in the deletion of said Item." (See page 3, paragraph [0028] "The secondary deletion ensures that a storage object and corresponding storage object entry are only removed if the storage object is not a member of any further groups after eliminating the relationship between the storage object and group i." This follows the request to delete the "group leader" which represents the group ID of the "storage group" or the "Item Folder" as in referred to in the claim.)

Claim 28:

Anglin discloses "wherein said Item is automatically deleted when it no longer belongs to any Item Folder." (See page 3, paragraph [0029] "If the target group is the only indicated group in the associated groups fields for the specified storage object entry, then the storage management server deletes the specified storage object entry from the storage database and deletes the identifier of the deleted storage object entry from the group entry for the target group.")

Claim 30:

Anglin discloses "wherein said Item is automatically deleted when it is a member of only one Item Folder and said Item Folder is deleted." (See page 3, paragraph [0028] "The secondary deletion ensure that a storage object and corresponding storage object entry are only removed if the storage object is not a member of any further groups after eliminating the relationship between the storage object and group i." This follows the request to delete the "group leader" which represents the group ID of the "storage group" or the "Item Folder" as in referred to in the claim.)

21. Claim 24 is rejected under 35 U.S.C. 103(a) as being unpatentable over US patent application 2004/0199521 by Anglin et. al. (hereafter Anglin), US Patent 7158962 by Nelson (hereafter Nelson), and and US Patent 6571245 by Huang et. al. (hereafter Huang) further

in view of "Typing a Multi-Language Intermediate Code" POPL 01' London, UK by Gordon et. al. (hereafter Gordon).

__Claim 24:__

Anglin, Nelson, and Huang do not explicitly disclose "The hardware/software interface system of claim 21 wherein said Item is a fundamental unit of information manipulated by a Common Language Runtime. " However, Gordon teaches said Item is a fundamental unit of information manipulated by a Common Language Runtime. (See conclusion p. 257 "One of the innovations in Microsoft's Common Language Runtime is support for typed stack pointers, for passing arguments and results by reference, for example. We presented formal typing rules and a type safety result for a substantial fragment of Common Language Runtime intermediate language. Our treatment of value types and pointer types appears to be new.") It would have been obvious to one with ordinary skill in the art to combine the teaching of Anglin, Nelson, and Huang with the disclosure of Gordon because of the motivation of being able to pass the arguments and results by reference to have more efficient processing of the data, less transfer overhead, and to be able to handle more robust types of objects. It is for this reason that one of ordinary skill in the art would have been motivated to have said Item is a fundamental unit of information manipulated by a Common Language Runtime.

22. Claim 25 is rejected under 35 U.S.C. 103(a) as being unpatentable over US patent application 2004/0199521 by Anglin et. al. (hereafter Anglin), US Patent 7158962 by Nelson

(hereafter Nelson), and US Patent 6571245 by Huang et. al. (hereafter Huang) further in view of US Patent 6430564 by Judge et. al. (hereafter Judge)

Claim 25:

Anglin, Nelson, and Huang do not explicitly disclose "The hardware/software interface system of claim 21 wherein said Item is a fundamental unit of information manipulated by a Java Virtual Machine". However, Judge teaches said Item is a fundamental unit of information manipulated by a Java Virtual Machine. (See abstract "A data manager manages global data within a Java Virtual Machine (JVM) installed and running in an embedded device. The data manager maintains a data class list that stores data class identifiers associated with each data class object currently loaded and cached in a data cache in the embedded device.") It would have been obvious to one with ordinary skill in the art to combine Anglin, Nelson, and Huang with Judge by using the JVM to allow for more diverse types of objects to be processed by the hardware/software interface system. It is for this reason that one of ordinary skill in the art would have been motivated to have said Item is a fundamental unit of information manipulated by a Java Virtual Machine.

23. Claims 29 and 31 is rejected under 35 U.S.C. 103(a) as being unpatentable over US patent application 2004/0199521 by Anglin et. al. (hereafter Anglin) and US Patent 7158962 by Nelson (hereafter Nelson), and US Patent 6571245 by Huang et. al. (hereafter Huang), US Patent 6438545 by Beauregard et. al. (hereafter Beauregard) further in view US 20040073560 by Edwards (hereafter Edwards)

Claim 29:

Anglin, Nelson, Huang, and Beauregard do not explicitly disclose "
when it no longer belongs to any Item Folder, automatically becomes a member of a default Item Folder." However, Edwards teaches said Item, when it no longer belongs to any Item Folder, automatically becomes a member of a default Item Folder. (See page 3, paragraph [0038] "The synchroniser can be set to 'Recycle' rather than delete files. This means that whenever the synchroniser is to over-write or delete a file, the file is passed to the operating system to be 'recycled'. This means it is not deleted immediately, but stored in an area where it can be retrieved if required." In the instant application, the area in which the item is stored is called the default Item Folder) It would have been obvious to one with ordinary skill in the art to combine the Item and Item folder system of Anglin, Nelson, Huang, and Beauregard with the disclosure of a recycling method of Edwards by simply adding the recycling method of Edwards to the system of Anglin, Nelson, and Beauregard. Edwards points out the advantage of keeping the potentially deleted file (or in the instant application, the object,) available for future use rather than immediate deletion. It is for this reason that one of ordinary skill in the art would have been motivated to have said Item, when it no longer belongs to any Item Folder, automatically becomes a member of a default Item Folder.

Claim 31:

Anglin, Nelson, Huang, and Beauregard do not explicitly disclose "said Item, when it is a member of only one Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder." However, Edwards teaches said Item, when it is a member of only one Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder. (See page 3, paragraph [0038] "The synchroniser can be set to 'Recycle' rather than delete files. This means that whenever the synchroniser is to over-write or delete a file, the file is passed to the operating system to be 'recycled'. This means it is not deleted immediately, but stored in an area where it can be retrieved if required." In the instant application, the area in which the item is stored is called the default Item Folder.) It would have been obvious to one with ordinary skill in the art to combine the Item and Item folder system of Anglin, Nelson, Huang, and Beauregard with the disclosure of a recycling method of Edwards by simply adding the recycling method of Edwards to the system of Anglin, Nelson, and Beauregard. Edwards points out the advantage of keeping the potentially deleted file (or in the instant application, the object,) available for future use rather than immediate deletion. It is for this reason that one of ordinary skill in the art would have been motivated to have said Item, when it is a member of only one Item Folder and said Item Folder is deleted, automatically becomes a member of a default Item Folder.

Response to Arguments

24. Applicant's arguments with respect to claim 2/15/08 have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

25. The prior art made of record listed on PTO-892 and not relied, if any, upon is considered pertinent to applicant's disclosure.

Contact Information

26. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Michael D. Pham whose telephone number is (571)272-3924. The examiner can normally be reached on Monday - Friday 9am - 5:00pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John R. Cottingham can be reached on 571-272-7079. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/M. D. P./
Examiner, Art Unit 2167

/John R. Cottingham/
Supervisory Patent Examiner, Art Unit 2167